Active Learning Strategies Based on Text Informativeness

Ruide Li
Kyoto University
Kyoto, Japan
li.ruide@dl.soc.i.kyoto-u.ac.jp

Yoko Yamakata
The University of Tokyo
Tokyo, Japan
yamakata@mi.u-tokyo.ac.jp

Keishi Tajima
Kyoto University
Kyoto, Japan
tajima@i.kyoto-u.ac.jp

***Abstract*—In this paper, we propose several strategies for selecting the next item to label in active learning for text data. Text data have several text-specific features, such as TF-IDF (term-frequency and inverse document frequency) vectors, and document embeddings based on the embeddings of the included words. These features have correlation with the informativeness of the text data, so they are potentially useful to decide annotation order in active learning. Our methods select the next item to label by using these text-specific features. We evaluate the performance of our strategies in two problem settings: the standard active learning setting, where we focus on the improvement of the model accuracy with as small number of annotations as possible, and the learning-to-enumerate setting, where we focus on the efficiency in enumerating all instances of a given target class with as small number of annotations to the non-target instances as possible. We also combine our strategies with two existing strategies: uncertainty sampling, a well-known strategy for the standard active learning, and the exploitation-only strategy, a strategy used in learning-to-enumerate problems. Our experiment on two publicly available English text datasets show that our method outperforms the baseline methods in both problem settings.**

***Index Terms*—active learning, learning to enumerate, informativeness, TF-IDF, word embedding, uncertainty sampling, exploitation-exploration**

I. Introduction

Sufficient size of labeled training dataset is crucial for success in supervised machine learning. However, human annotation is often cost-prohibitive in terms of time and money. To accomplish a task with as few human annotations as possible, it is desirable if we can assign labels only to items that are most effective as training data. However, we cannot know which items are effective in advance. To tackle this problem, many studies have proposed various strategies for actively (i.e., not passively) deciding the next data items to label and learn. This approach is called active learning [1].

In active learning, we train a classifier and create a training dataset incrementally in parallel. We use the current classifier (initially a random classifier) to decide which item to label next, the obtained label is added to the training dataset, and we update the classifier by using the updated dataset. We then use the updated classifier for selecting the next item. We repeat these steps in order to train the classifier efficiently with a minimum annotation effort. We usually assume that the labels given to the chosen items are always correct, and they are sometimes called oracles.

There are several variations of the scenario assumed in active learning, but we focus on the scenario called pool-based active learning [1]. In pool-based active learning, we are given a fixed pool of unlabeled data, and the next data item to label is chosen from the pool. In this scenario, what we need is a strategy for selecting the next item from a given set.

There are several criteria that are important for item selection in active learning. One such criterion is informativeness of an item and its label. There have been many studies on how to define and estimate the informativeness in the active learning context. Most of the proposed methods have been designed for general applications on arbitrary data types. However, when we apply active learning techniques to some specific task, the task usually processes some specific types of data. Therefore, we may be able to make use of some properties specific to the data types for estimating the informativeness of items.

Text data is one of the data types that most frequently appear in machine learning tasks. In this paper, we propose several strategies for selecting the next item in active learning on text data. Our strategies use features that are specific to text data. One of the most important characteristics of text data is that a text is composed of words. We hypothesize that a text data including more informative words contributes more to improving the classifier. We propose methods for estimating the informativeness of a text data by using the following word-related text-specific features: unique word count, TF-IDF (term frequency and inverse document frequency) values of the included words, and document embeddings based on the embeddings of the included words.

When we evaluate the proposed methods, we consider two variations of the problem setting in active learning. The first is the standard one, in which the goal is to achieve high model accuracy with as few annotations to data items as possible. The second is the problem setting called learning-to-enumerate problem [2], in which the goal is to enumerate all the instances of a given target class with as few annotations to items that are not instances of the target class. We explain the difference between these two problem settings in more detail later.

Informativeness estimated from text-specific features and other data-type-independent criteria used in the existing studies focus on completely different aspects of data items. We, therefore, expect that they may be complementary with each other. Based on this intuition, we also propose methods that

combine our methods and existing standard methods for active learning. In particular, we combine our methods with uncertainty sampling [3] for the standard active learning setting, and we combine our method with the greedy exploitation-only strategy for the learning-to-enumerate setting.

We conducted experiments with two publicly available English text datasets, and the results suggest that our methods have some superiority over the existing methods in the two problem settings explained above.

The remainder of this paper is organized as follows. Section II discusses some related work. Section III explains the two problem settings in more details. Section IV explains our proposed method, and Section V shows the result of our experiments. Section VI concludes the paper.

II. Related Work

In this section, we first explain existing research on the active learning and some other problems related to the training order of samples in machine learning. We then briefly survey the research on active learning for text data. We also explain related work on the informativeness of words and documents.

A. Training Order of Samples in Machine Learning

In active learning, we repeatedly choose an unlabeled item, query its label, and feed the result into the classifier for training. Settles [1] further classified this scenario into three types of scenarios: pool-based, stream-based, and membership querying. In pool-based active learning, we choose the next item from a given pool of unlabeled data. In stream-based active learning, we are given an item one-by-one, and we need to decide whether we query its label or not before proceeding to the next item. In other words, pool-based active learning is an offline problem, and stream-based active learning is an online version of the problem. On the other hand, some studies have assumed that we can generate a sample data and query its label. This scenario is called membership querying.

In this paper, we focus on pool-based active learning. In many applications of information retrieval, we are given a fixed set of data items. In the learning-to-enumerate setting, we are always given a fixed pool of unlabeled data items. In addition, we focus on text data in this paper. When we focus on text data, membership querying scenario is not common because it is not easy to artificially generate useful sample documents, and it may also be difficult for human annotators to label such artificially generated documents appropriately.

The main stream of the research on active learning do not assume specific data types, and they mainly use information on the distribution of the data in the data space. Uncertainty sampling proposed by Lewis and Gale [3] is one such method. It chooses the item for which the current model has the least confidence (i.e., an item closest to the decision boundary of the current model). The intuition behind this is that the correct label for such an item must be the most informative for the model, and will best improve the model. On the contrary, an item for which the current model has the highest confidence about the class must be the least informative. Their experiment on a text categorization task showed that uncertain sampling could achieve reductions of up-to 500-fold in the number of annotations [3]. Because uncertainty sampling has been proved to be an effective and also robust method [4], [5], we adopt it as the standard method in the active learning setting, and combine our method with it. We use uncertain sampling also as the baseline method in the evaluation explained in Section V.

A similar method of estimating the informativeness of items is query-by-committee [6], [7], where we train multiple classifiers, and choose items on which they disagree.

On the contrary, in relevance feedback [8], which is a well-known technique in information retrieval, we show to users items top-ranked by the current ranking model, and ask for their feedback (i.e., relevance labels) on them. The ranking model is then updated based on the feedback. In other words, we query for labels of items that the current model thinks is the most likely to be of the target class.

It is a reasonable strategy when the annotators are the querying users themselves who are interested only in reading documents in the target class. This strategy is also expected to improve the classifier well if many negative samples are included in the top-ranked answers by the current model. It is also reasonable when we assume interactive information retrieval tasks. In that case, if all the top-ranked answers are of the target class, we can finish the task, and if there are negative samples within the top-ranked answers by the current ranking model, we can improve the model.

In ordinary active learning, however, we repeat the labeling step many times, and as the training proceeds, negative samples become very rare in the top-ranked answers. Therefore, if we only label top-ranked items in the ordinary active learning scenario, we cannot improve the classifier until we run out of positive unlabeled samples in the top-ranked answers.

In the learning-to-enumerate setting, the problem explained above is not an issue because the goal is to extract the positive instances until we run out of it. The main issue in the learning-to-enumerate setting is how to balance exploitation and exploration. In this context, exploitation means that we choose an item that the current model thinks is the most likely to be of the target class in order to maximize the probability that we obtain a positive instance. Exploration means that we choose an item for which the current model is least confident in order to explore a new region in the data space.

Jörger et al. [2] discussed the learning-to-enumerate problem and proposed a simple method based on ϵ-greedy strategy [9]. That is, their method adopts the selection by the current classifier in the probability $1 - \epsilon$, and explore items in new regions in the data space in the probability ϵ. They experimented on 19 small- and medium-sized public datasets available at the UCI Machine Learning Repository, and their results showed that the exploitation-only strategy (i.e., $\epsilon = 0$) was the best. Based on their result, we combine our methods with the exploitation-only strategy when we assume the learning-to-enumerate setting. The exploitation-only strategy is also used as the baseline when we evaluate our methods in the learning-to-enumerate setting.

We explained that there is an exploitation–exploration trade-off in the learning-to-enumerate problem, but standard active learning also has a trade-off between refining the current decision boundary and exploring unexplored regions. Osugi et al. [10] proposed an active-learning algorithm that dynamically adjust the probability of exploration at each step based on the effectiveness of the previous exploration. The effectiveness of the previous exploration is measured by how much it has changed the model. This kind of methods that balance exploitation and exploration are orthogonal to methods that either implement exploitation strategy or implement exploration strategy. For example, their method can be combined with uncertainty sampling: we use uncertainty sampling when their method has selected exploration, and use some exploitation strategy when their method has selected exploitation.

Konyushkova and Raphael [11] proposed another approach, which they called "learning active learning." Their idea was to train a regressor that predicts the expected error reduction we would obtain by learning each candidate sample. The input to the regressor are the properties of the current classifier and the candidate sample. They first trained a random-forest classifier on a synthetic labeled dataset and measured the error reduction by each sample. They then trained a regressor that predicts the error reduction based on the properties of the current classifier and the sample. They showed that it can learn a strategy that works well on real data from a wide range of domains. However, we cannot use their method in our problem setting because their method requires the labeled synthetic data for training the regressor that predicts the error reduction.

In addition to informativeness, representativeness is an important criteria in the item selection in active learning [12], [13]. Items that are representative of the remaining unlabeled data are more useful for improving the classifier accuracy on those remaining data. Representativeness is important when the distribution of the remaining unlabeled data is skewed. Zhu et al. [14] proposed an active learning strategy that consider both uncertainty and representativeness of items, and applied it for active learning on text data. They measure the representativeness of an item by the density of other items in its neighbor in the data space.

Bengio et al. [15] proposed the concept of curriculum learning, which also manages the training order in supervised machine learning. They trained a classifier starting with simpler data (e.g., document with less vocabulary), then gradually proceeded to more complex ones (e.g., documents with more vocabulary). Their experiment showed that this strategy achieved faster convergence and higher final classifier accuracy. They assume that we use all the training data, and consider the learning order of them. On the other hand, in the active learning and the learning-to-enumerate problem, we consider what to label next in order to avoid labeling all items.

B. Active Learning for Text Data

There have been studies on active learning for tasks related to text data, such as word sense disambiguation [14], [16] and word segmentation [17]. For example, Chen et al. [16] analyzed what features are useful for word sense disambiguation task in the active learning scenario. However, these studies do not use text-specific properties for selecting next items.

There have also been many studies that use feature vectors produced by deep neural networks (DNNs) from text data for active learning [18]–[21]. For example, An et al. [20] used feature vectors extracted by Recurrent Neutral Network (RNN) for active learning on text data. Kholghi et al. [21] also confirmed that features produced by word embeddings are useful for active learning on text data. However, these studies do not use these feature for selecting the next item.

Recently, some studies proposed active learning strategies for training deep neural networks (DNNs) for text classification tasks [22], [23]. However, in active learning, we update the classifier many times (more than 3,000 times in one of our experiments explained in Section V). Computation resources for such a computation is not always available, and some studies use classifiers with lower complexity [19], [24], [25]. In addition, DNNs require a larger training datasets, and we often want to achieve a reasonable performance with a smaller number of annotations. For these reasons, we use linear support vector machines (linear SVMs) instead of DNNs.

C. Informativeness of Words and Documents

The most classic and well-known measure of the informativeness of words is the inverse document frequency (IDF) [26], [27]. There have been many proposals of other measures of the informativeness of words including z-measure [28], Residual IDF [29], Gain [27], and clarity [30], but IDF is still regarded as the best measure and the most popularly used. For example, Rennie and Jaakkola [31] proposed a new measure, and experimentally compared the performance of IDF, their measure, and several other measures in their task of named entity detection. The result shows that their measure can complement IDF, but IDF still outperforms their measure when each of them is used alone.

There have also been studies on informativeness of documents [32], [33], but they focus on the usefulness of a document for the readers, not the informativeness for a classifier.

III. Two Problem Settings

As explained before, we evaluate our proposed methods and the baseline methods in two problem settings: the standard active learning setting and the learning-to-enumerate setting.

The learning-to-enumerate setting focuses on how efficiently we can extract all instances of a specific class from a fixed pool of unlabeled data items. In some applications, we want to find instances of a specific target class as promptly as possible while we train a model in parallel.

For example, after a natural disaster, we want to find all tweets asking for help as promptly as possible. Because we cannot examine all tweets posted from the affected area, we want to adopt some machine learning techniques. However, we cannot prepare a trained classifier in advance because keywords included in messages asking for help differ from a disaster to a disaster depending on the disaster type and the

area. Therefore, we need to adopt active learning. We select tweets that are the most likely to be asking for help, and ask human workers to examine if they really are. We use the results not only for deciding where to send the rescue teams, but also for training a classifier, which is then used to select the next tweets to examine. Because human examination takes some time, our goal is to find all tweets asking for help by examining the smallest number of tweets.

The learning-to-enumerate setting is different from the standard active learning setting. In the standard active learning, we choose the item that would best improve the classifier, no matter what the label (i.e., the result of the oracle or human examiner) of the item would be. On the other hand, in the learning-to-enumerate problem, our goal is to minimize the number of items we manually examine before we find all the instances of the target class. It is equivalent to minimize the number of "misses", that is, the number of items that are examined and turned out not to be the target class. It means that labeling an item in the target class is not regarded as a cost. Therefore, items that are more likely to be of the target class are preferred when we select the next item to label.

One of the standard approach in the standard active learning problem setting is uncertainty sampling [3], as explained in Section II. In uncertainty sampling, we choose an item for which the current model of the classifier is most uncertain.

On the other hand, in the learning-to-enumerate setting, we may choose and label an item which the current model thinks is the most likely to be of the target class, because labeling an item of the target class is not the cost. There is, of course, a trade-off between exploitation and exploration. If we only choose such items, the model will not be improved, and we cannot obtain a model as good as a model we could have obtained if we adopted uncertainty sampling. However, according to [2], the exploitation-only strategy is the best for learning-to-enumerate problems in most cases in their extensive experiments, as explained in Section II. In other words, when we select the next item to label in the learning-to-enumerate problem, we should choose an item that the current model thinks is the most likely to be of the target class.

In this way, the standard problem setting of active learning and that of the learning-to-enumerate problem have different solutions. In the standard active learning, we should choose an item that is most informative to the classifier, and in the learning-to-enumerate problem, we should consider the balance of the exploitation and exploration. In this paper, we evaluate our proposed methods in the both problem settings.

Although the learning-to-enumerate setting frequently appears in many applications, it has not been studied sufficiently. To emphasize that this setting is not uncommon, we show one more example. Suppose we are given a news archive, and want to retrieve all the articles published between 1918 and 1920, and related to Spanish flu. We may not have a classifier or a training dataset for such a topic in advance because it was not a important topic until we had COVID-19. If we have neither a classifier nor a training dataset, it is another example of the text retrieval task in the learning-to-enumerate setting.

IV. Proposed Method

We first propose several primitive methods that select the most informative document based on some text-specific features. We then explain our combined methods that combine one of the primitive methods and either uncertainty sampling or the exploitation-only strategy.

A. Primitive Methods

Unique Word Count. The first method sort the documents by unique word count, i.e., the number of words in a document without counting multiple occurrences of the same word. The intuition behind this method is that documents having more unique words must be more informative for training a classifier. In our problem setting, we assume that the cost for labeling one document is constant no matter how long the document is and how many words it includes. Therefore, if we label a document including more words, we pay a constant cost and obtain more information on which words are positive/negative supports of the target class.

Sum of TF-IDF. Our first method uses unique word count. However, multiple occurrences of some words may also be informative for the classifier. If some words appears many times in a document, we can expect that the document has a very specific topic, and such a document must be more informative for the classifier than a document that has no prominent word. This argument, of course, does not hold for very common words that appear many times in all or many documents. Based on these observations, our second method computes the score of a document by summing the TF-IDF values of the words in it.

However, in our experiment, we found out that the score calculated by this method is too much affected by very unusual words (that is, very large IDF values) that happen to appear in a document only once. In a extreme case, if a very unusual word appears only in that document and does not appear in any other documents in the given pool, information on that word is not useful for the classifier. This is the issue related to the representativeness explained in Section II. To avoid this problem, we only use words having top-k TF-IDF scores in the document. When a document includes many rare words that appear in the document only once, the sum of their TF-IDF values can largely affect the score, but such words are excluded when we select words with top-k TF-IDF values because a document usually has more than k words with large enough TF values. In other words, we only use such ordinary words and exclude unusual words.

Sum of TF-IDF of Unseen Words. Even if a word is informative, if the classifier has already learned that word, it is no longer useful for improving the classifier. Following this observation, the third method only select words that have not appeared in the already-labeled data and sums-up their TF-IDF values. This method tries to select documents that include many unseen informative words that have never appeared in the documents we have already labeled.

Norm of Embedding Vector. Word embedding, which transforms a word into a vector, has recently been proven

very useful. Word2vec [34] is one of the classic methods for word embedding. Schakel et al. [35] discussed the relationship between the norm of a Word2vec vector and the significance of the word. According to their experiment, the norm becomes larger when the frequency of the word increases to a given point. If the frequency further grows, it means that the word can appear in various contexts, and the norm becomes smaller. Their finding suggests that the norm of embedding vectors is useful for estimating informativeness of words. Doc2vec [36] is a methods for transforming a whole document into a vector based on the same concept as Word2vec. Our fourth method transforms documents into vectors by using Doc2vec and sort documents by the norm of the obtained vectors.

There have been proposals of other more successful word embedding techniques, such as BERT [37] and RoBERTa [38]. In this paper, however, we adopted Doc2vec because it has been confirmed by [35] that the Word2vec vectors have the properties explained above, and Doc2vec is expected to share the same property.

B. Combined Methods

We also consider combinations of our primitive methods and uncertainty sampling or the exploitation-only strategy. In the standard active learning setting, we use uncertainty sampling, and in the learning-to-enumerate setting, we use the exploitation-only strategy. We first select the top-l items by using uncertainty sampling or the exploitation-only strategy, then we select one of them by using one of our primitive methods. The intuition behind this design is as follows.

Uncertainty sampling has been proven to work very well, but it sometimes fail, and one of the reasons of the failures is that uncertainty sampling sometimes selects unusual outliers without meaningful contents. To avoid that, our combination methods select an item that seems most meaningful among the top-l items chosen by uncertainty sampling.

In the learning-to-enumerate setting, our combined methods first select top-l candidates which the current model thinks are the most likely to be of the target class. In most cases, all of these items are truly the positive instances. Therefore, we can quite safely choose any of them. If so, we should choose the most informative one among them.

V. Experiments

In our experiment, we start from the situation where one positive and one negative samples are given. If there are no such samples in the actual application, we randomly select items from the pool until we find them. Below are the values we used for parameters in our methods.

- When selecting top-k words having the highest TF-IDF values in our method, we selected 20 words, i.e., $k = 20$.
- In the combination methods, we first choose top 10 candidates, i.e., $l = 10$.

For the machine-learning model, we selected a linear support vector machine (SVM) with all default hyper-parameters from the SciKit-Learn library [39], and we used the balanced class weight for training, because the datasets were initially unbalanced. We also tested random forest classifier, but it performed poorly. We did not use DNNs because of their huge computation cost as explained in Section II.

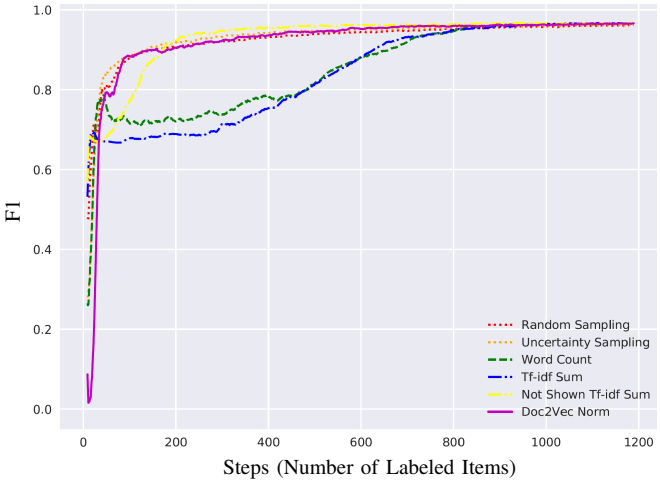
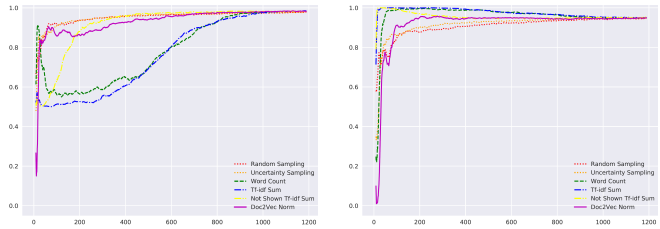


Fig. 1. F1 Score on Dataset 1



(a) precision (b) recall

Fig. 2. Precision/Recall on Dataset 1

A. Datasets

We used two datasets: Dataset 1 and Dataset 2. Dataset 1 is the Short-Message Service (SMS) Spam Collection Dataset available at UCI Machine Learning Repository [40]. It is a set of SMS messages labeled as "spam" or "ham." In the learning-to-enumerate setting, we need to specify a target class. For this dataset, we define spam as the target class. Dataset 2 is the Large Movie Review Dataset v1.0 [41]. This is a dataset for binary sentiment classification of movie reviews. For this one, we define positive reviews as the target class in the learning-to-enumerate setting.

To investigate the effect of data balance, we manually changed the target data ratio in the two datasets. In this paper, we report the results where the target data ratio in Dataset 1 and in Dataset2 was set to 50% and 20%, respectively.

B. Results for Dataset 1

We measured F1 score at each step of the annotation process, that is, after labeling each item and re-train the classifier. In the following, we plot the results in graphs where x-axes represent the number of labeled data. All curves are smoothed by a moving average with window size of 10 so that the figures can be read more clearly.

Figure 1 shows F1 score of the proposed methods and two baseline methods, uncertain sampling and random sampling,

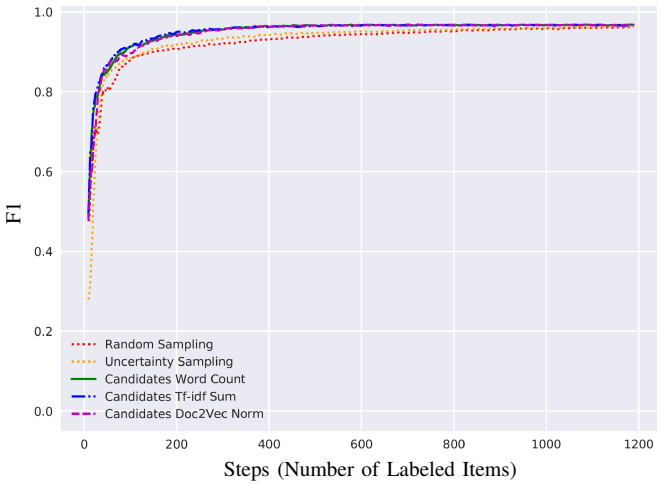


Fig. 3. F1 score of combined methods in Active Learning on Dataset 1

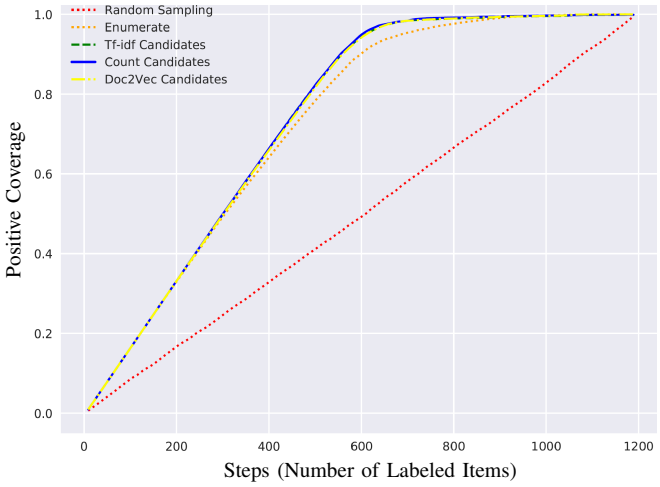


Fig. 4. Recall in Learning-to-Enumerate Setting on Dataset 1

on Dataset 1. In the very beginning, our TF-IDF method and our unseen TF-IDF method are the best, but soon their F1 scores stop to increase. After that, random sampling becomes the best, and then our word count method, uncertainty sampling, our Doc2Vec method, and uncertainty sampling again, become the best. Finally our unseen TF-IDF method becomes the best method and keep it for many steps.

The goal of the active learning in the standard setting is to promptly achieve the performance we would finally obtain when we train the classifier with a sufficiently large dataset. In that sense, our unseen TF-IDF method is the best in this experiment, but the second best is uncertainty sampling, and the margin is very small. In addition, if we compare the area under the curve (AUC) in this graph, uncertainty sampling is the best, although high AUC is not the goal of the active learning as explained above.

To analyze more details of the result, we also show the graphs for precision and recall in Figure 2. As shown in these graphs, our methods tend to have lower precision and higher recall than uncertainty sampling. This is because items that are given higher scores by our methods are more likely to be

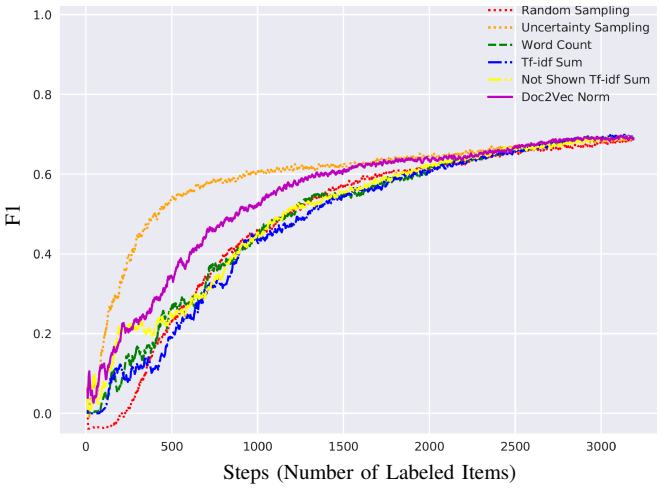
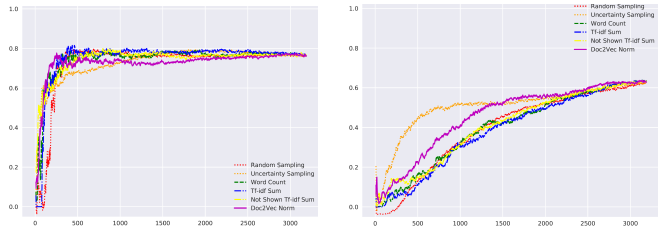


Fig. 5. F1 Score in Active Learning on Dataset 2



(a) precision (b) recall

Fig. 6. Precision/Recall on Dataset 2

of the target class, i.e., spams, in this dataset. As a result, the items chosen and labeled earlier are biased toward to spams, and the classifier trained by those items is also biased toward the spam class. However, because this dataset is balanced, i.e., including 50% spam and 50% ham, this bias is not the reason of the superiority or inferiority of the proposed methods.

We also evaluate our methods combined with uncertainty sampling, i.e., the methods that first select top-10 items by uncertainty sampling, and select one item from those candidates by using one of our methods. Figure 3 shows F1 score of the methods. As shown in this graph, the methods that combine our methods and uncertainty sampling showed slightly but consistently better performance than did simple uncertainty sampling. The difference between our methods are very small.

Next, we evaluate our methods in the learning-to-enumerate setting. In the learning-to-enumerate setting, the goal is to find instances of the target class as promptly as possible. Therefore, we evaluate the methods by recall, i.e., the ratio of the labeled and found positive instances to all the target instances in the pool. To distinguish it from recall in the experiments in the standard active learning setting, we call it positive coverage.

In this setting, our primitive methods not combined with the exploitation-only strategy did not perform well, so we omit their result. Figure 4 shows positive coverage of the exploitation-only strategy (denoted by Enumerate in the graph) and also positive coverage of our methods combined with it. The combined methods showed slightly but consistently better performance than did exploitation-only method. Differences

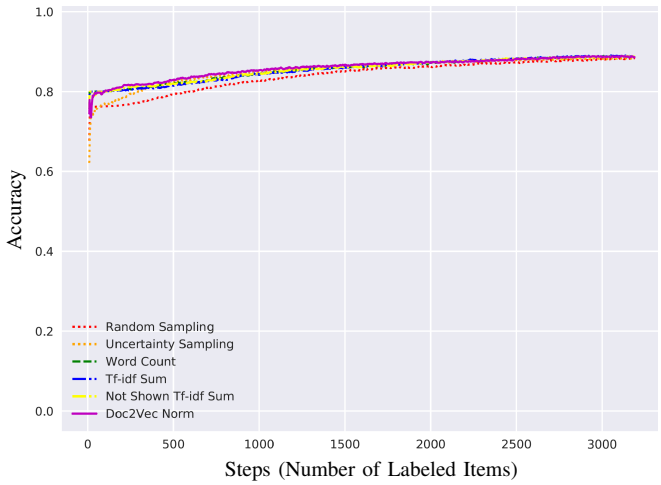


Fig. 7. Accuracy in Active Learning on Dataset 2

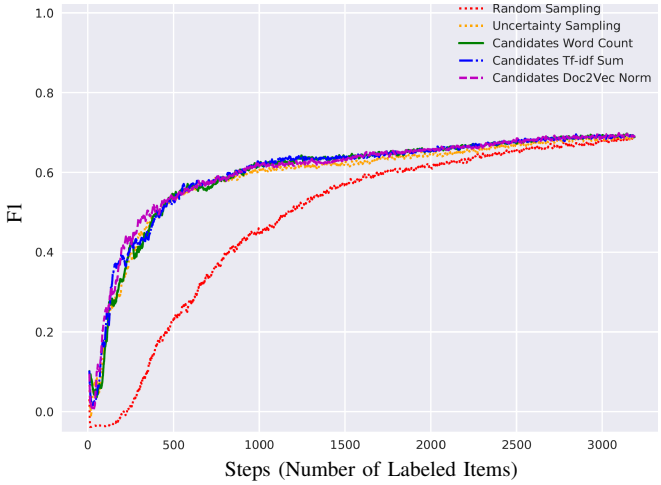


Fig. 8. F1 Score of Combined Methods in Active Learning on Dataset 2

among the proposed methods were very small.

However, only from this result, we cannot know if our method is really superior. In this dataset, the positive instances tend to be given higher scores by our method as explained before, and it can be the reason of this result. If it is the reason, our method would be outperformed by the exploitation-only strategy when the target class is the opposite, i.e., "hams". For Dataset 2, we will have such an opposite case, so we will draw a conclusion after explaining the result for Dataset 2.

C. Results for Dataset 2

We next show the result of the experiment with Dataset 2. Notice that Dataset 1 comprised 50–50% balanced data, whereas Dataset 2 was 20–80% unbalanced.

Figure 5 shows F1 scores of our methods, uncertain sampling, and random sampling. For this dataset, our methods were outperformed by uncertainty sampling at most steps. To analyze further, we show precision and recall in Figure 6. What happened here is just the opposite to what happened for Dataset 1. In this dataset, items that are given higher scores by our methods are more likely to be of the non-target

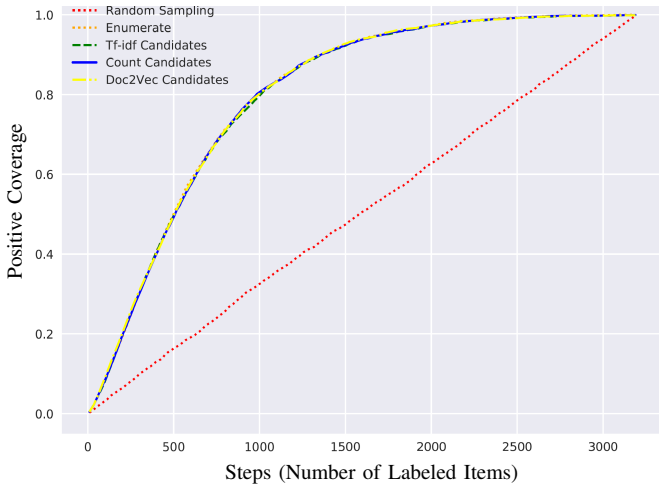


Fig. 9. Recall in Learning-to-Enumerate Setting on Dataset 2

class, i.e., negative reviews. Because of that, item selection by our methods are biased toward the non-target classes, and the classifiers trained by them are biased toward non-target classes. As a result, they have higher precision and lower recall than uncertainty sampling.

Because our methods are biased toward the non-target class, and this dataset is 20%-80% dataset including more non-target instances, if we compare our methods and uncertainty sampling based on accuracy instead of F1 score, our methods outperforms uncertainty sampling as shown in Figure 7.

Next, we compare uncertainty sampling and our methods combined with it. Figure 8 shows their F1 scores. All methods are close to each other, but our Doc2Vec method outperformed uncertainty sampling with a narrow but consistent margin.

Finally, we evaluate our methods in the learning-to-enumerate setting with Dataset 2. Our primitive methods not combined with the exploitation-only strategy did not perform well also for Dataset 2, so we omit their result, and only show the result of our methods combined with the exploitation-only strategy. Figure 9 shows the positive coverage of the exploitation-only strategy (denoted by Enumerate) and our methods combined with it. For Dataset 2, instances of the non-target classes, i.e., negative reviews, are more likely to be given higher scores by our method, as explained above. It is disadvantageous for our method in the learning-to-enumerate setting. However, despite that disadvantage, the performance of our methods and the exploitation-only strategy are almost equal. Therefore, this result and the result for Dataset 1 shown in Figure 4 suggest that our methods combined with the exploitation-only strategy have superiority over the simple exploitation-only strategy.

VI. Conclusion

In this research, we focused on active learning for text classification tasks. We considered two variations of the problem setting of active learning: the standard active learning setting and the learning-to-enumerate setting.

Most existing strategies for item selection in active learning do not assume specific data types, and designed for arbitrary data types. On the other hand, we proposed several methods that utilize features specific to text data. Our methods estimates the informativeness of text data by using unique word counts, sums of TF-IDF values of all words, sums of TF-IDF values of unseen words, and the norms of Doc2vec vectors.

We also proposed methods combining these primitive methods with existing standard methods. For standard active-learning setting, we combine our methods with uncertain sampling. For the learning-to-enumerate setting, we combine our methods with the greedy exploitation-only strategy.

We conducted experiments comparing these proposed and existing methods on two datasets. In the standard active learning setting, our primitive methods did not necessarily outperform uncertainty sampling, but our combined methods outperformed it with a small but consistent margin.

In the learning-to-enumerate setting, our methods outperformed the exploitation-only strategy in the experiment with Dataset 1, where our methods have an advantage because of the property of the target class, and our methods showed performance almost equal to that of the exploitation-only strategy in the experiment with Dataset 2, where our methods have disadvantage because of the property of the target class. These results suggest that our methods have superiority over the simple exploitation-only strategy.

REFERENCES

[1] B. Settles, "Active learning literature survey," University of Wisconsin–Madison, Techinical Report 1648, 2009.

[2] P. Jörger, Y. Baba, and H. Kashima, "Learning to enumerate," in *Proc. of Intl. Conf. on Artificial Neural Networks, Part I*, 2016, pp. 453–460.

[3] D. D. Lewis and W. A. Gale, "A sequential algorithm for training text classifiers," in *Proc. of SIGIR*, 1994, pp. 3–12.

[4] D. D. Lewis and J. Catlett, "Heterogeneous uncertainty sampling for supervised learning," in *Proc. of ICML*, 1994, pp. 148–156.

[5] M.-F. Balcan, A. Broder, and T. Zhang, "Margin based active learning," in *Proc. of Intl. Conf. on Computational Learning Theory*. Springer, 2007, pp. 35–50.

[6] H. S. Seung, M. Opper, and H. Sompolinsky, "Query by committee," in *Proc. of the Annual Workshop on Computational Learning Theory*, 1992, pp. 287–294.

[7] I. Dagan and S. P. Engelson, "Committee-based sampling for training probabilistic classifiers," in *Proc. of ICML*, 1995, pp. 150–157.

[8] G. Salton and C. Buckley, "Improving retrieval performance by relevance feedback," *JASIS*, vol. 41, no. 4, pp. 288–297, 1990.

[9] C. Watkins, "Learning from delayed rewards," Ph.D. dissertation, University of Cambridge, Cambridge, England, 1989.

[10] T. T. Osugi, K. Deng, and S. D. Scott, "Balancing exploration and exploitation: A new algorithm for active machine learning," in *Proc. of ICDM*, 2005, pp. 330–337.

[11] K. Konyushkova, R. Sznitman, and P. Fua, "Learning active learning from data," in *Annual Conference on Neural Information Processing Systems*, 2017, pp. 4225–4235.

[12] H. T. Nguyen and A. Smeulders, "Active learning using pre-clustering," in *Proc. of ICML*, 2004, p. 79.

[13] S. Dasgupta and D. Hsu, "Hierarchical sampling for active learning," in *Proc. of ICML*, 2008, pp. 208–215.

[14] J. Zhu, H. Wang, T. Yao, and B. K. Tsou, "Active learning with sampling by uncertainty and density for word sense disambiguation and text classification," in *Proc. of COLING*, 2008, pp. 1137–1144.

[15] Y. Bengio, J. Louradour, R. Collobert, and J. Weston, "Curriculum learning," in *Proc. of ICML*, 2009, pp. 41–48.

[16] J. Chen, A. Schein, L. Ungar, and M. Palmer, "An empirical study of the behavior of active learning for word sense disambiguation," in *Proc. of the Human Language Technology Conference of the NAACL, Main Conference*, 2006, pp. 120–127.

[17] M. Sassano, "An empirical study of active learning with support vector machines for Japanese word segmentation," in *Proc. of ACL*, 2002, pp. 505–512.

[18] C. Schröder and A. Niekler, "A survey of active learning for text classification using deep neural networks," *arXiv:2008.07267*, 2020.

[19] J. Lu, M. Henchion, and B. Mac Namee, "Investigating the effectiveness of representations based on word-embeddings in active learning for labelling text datasets," *arXiv preprint arXiv:1910.03505*, 2019.

[20] B. An, W. Wu, and H. Han, "Deep active learning for text classification," in *Proc. of ICVISP*, 2018, pp. 1–6.

[21] M. Kholghi, L. D. Vine, L. Sitbon, G. Zuccon, and A. N. Nguyen, "The benefits of word embeddings features for active learning in clinical information extraction," in *Proc. of the Australasian Language Technology Association Workshop*, 2016, pp. 25–34.

[22] Y. Zhang, M. Lease, and B. Wallace, "Active discriminative text representation learning," in *Proc. of the AAAI Conf.*, vol. 31, no. 1, 2017.

[23] Q. Liu, Y. Zhu, Z. Liu, Y. Zhang, and S. Wu, "Deep active learning for text classification with diverse interpretations," in *Proc. of CIKM*, 2021, pp. 3263–3267.

[24] G. Singh, J. Thomas, and J. Shawe-Taylor, "Improving active learning in systematic reviews," *arXiv preprint arXiv:1801.09496*, 2018.

[25] K. Hashimoto, G. Kontonatsios, M. Miwa, and S. Ananiadou, "Topic detection using paragraph vectors to support active learning in systematic reviews," *Journal of Biomedical Informatics*, vol. 62, pp. 59–65, 2016.

[26] K. S. Jones, "Index term weighting," *Information Storage and Retrieval*, vol. 9, no. 11, pp. 619–633, 1973.

[27] K. Papineni, "Why inverse document frequency?" in *Second Meeting of the North American Chapter of the ACL*, 2001.

[28] S. P. Harter, "A probabilistic approach to automatic keyword indexing. Part I. On the distribution of specialty words in a technical literature," *JASIST*, vol. 26, no. 4, pp. 197–206, 1975.

[29] K. W. Church and W. A. Gale, "Inverse document frequency (IDF): A measure of deviations from poisson," in *Third Workshop on Very Large Corpora, VLC@ACL*, 1995.

[30] S. Cronen-Townsend, Y. Zhou, and W. B. Croft, "Predicting query performance," in *Proc. of SIGIR*, 2002, pp. 299–306.

[31] J. D. Rennie and T. Jaakkola, "Using term informativeness for named entity detection," in *Proc. of SIGIR*, 2005, pp. 353–360.

[32] C. Horn, A. Zhila, A. Gelbukh, R. Kern, and E. Lex, "Using factual density to measure informativeness of web documents," in *Proc. ofNODALIDA*, 2013, pp. 227–238.

[33] N. Khairova, W. Lewoniewski, K. Wecel, M. Orken, and M. Kuralai, "Comparative analysis of the informativeness and encyclopedic style of the popular web information sources," in *Proc. of BIS*. Springer, 2018, pp. 333–344.

[34] T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Efficient estimation of word representations in vector space," *arXiv preprint arXiv:1301.3781*, 2013.

[35] A. M. J. Schakel and B. J. Wilson, "Measuring word significance using distributed representations of words," *CoRR*, vol. abs/1508.02297, 2015. [Online]. Available: http://arxiv.org/abs/1508.02297

[36] Q. V. Le and T. Mikolov, "Distributed representations of sentences and documents," in *Proc. of ICML*, 2014, pp. 1188–1196.

[37] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," 2018.

[38] Y. Liu, M. Ott, N. Goyal, J. Du, M. Joshi, D. Chen, O. Levy, M. Lewis, L. Zettlemoyer, and V. Stoyanov, "Roberta: A robustly optimized bert pretraining approach," *arXiv preprint arXiv:1907.11692*, 2019.

[39] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay, "Scikit-learn: Machine learning in python," *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.

[40] T. A. Almeida, J. M. G. Hidalgo, and A. Yamakami, "Contributions to the study of SMS spam filtering: new collection and results," in *Proc. of DocEng*, 2011, pp. 259–262.

[41] A. L. Maas, R. E. Daly, P. T. Pham, D. Huang, A. Y. Ng, and C. Potts, "Learning word vectors for sentiment analysis," in *Proc. of Annual Meeting of the ACL: Human Language Technologies - Volume 1*, 2011, pp. 142–150.